Compute Eigen Values

In Parallel Using openMP

Problem Statement

- What are Eigen Values?
- They are roots of the characteristic polynomial for a matrix.
- How to compute?
- Can find the characteristic polynomial and use the best way to compute roots of that equation.
- So Naive?
- Yes, need to change..!!

How to find?

- Can use QR decomposition method to find Eigen Values
- How?
- Small trick… Eigen values of two similar matrices are same.
- What are Similar Matrices?

Similar Matrices

- **Definition : A** and **B** are similar matrices if and only if there exists a non-singular matrix **P** such that $B=P^{-1}AP$. (or $PBP^{-1} = A$)
- **Theorm :** Similar Matrices have the same set of Eigen Values.

QR Algorithm

- Given a square matrix A we can factor A=QR where Q is orthogonal and R is upper triangular.
- **Algorithm:**
- **Start:** 1) $A_0 = A = QR$, 2) find $A_1 = RQ$ $\Leftrightarrow$ A_0 and A_1 are similar matrices
- **Intermediate:** We will continue this process until A_n, such that A_n is an upper triangular matrix.
- **Final:** Eigen values of A are same as diagonal elements of A_n.
- WHY?

WHY?

- Here we used another theorm:-
- The eigen values of an upper triangular matrix are nothing but its diagonal elements.
- The proof is not clearly discussed but it came from the truth that the product of eigen values of any matrix is the same as the product of all diagonal elements of that matrix.

What Next?

- This is a sequential way. But is that out target? No!!
- Need to parallelise it.
- Before that, let’s discuss the procedure of finding the QR decomposition.
- There are many ways to achieve this like Gram Scmidt process, House Holder Reflections, Givens Rotations etc
- But we chose to use Givens Rotation since it can be better parallelised.
- Now here is where we come about the parallelisation.
- How?

How to parallelise QR decomposition?

- On our first glimpse we may come to a conclusion that the givens rotations can’t be parallelized.
- Because for calculating a givens rotation the previous rotations need to be applied first, which make the algorithm hard to parallelize.
- However there do exist a method.
- ??

How to parallelise QR decomposition?

- Basically the matrix is split up into groups columnwise.
- Each processor calculates the rotations for its group of columns and passes the results onto the other processors.
- They apply the received rotations to their elements.
- The parallelization is designed such that the order will stay consistent.
- One main problem is that, this parallelisation needs a lot of communication.

Can we improve?

- It can be observed that the performance of QR iteration mainly depends on the number of non-zero entries below the diagonal.
- So it is helpful if we reduce the matrix to a compact form where this number is low.
- So our target is to reduce the given matrix to a compact form which is a similar matrix to A.
- These type of transformations are called similarity transformation which are of general form

$$A = S * A * S^{-1}$$

How to reduce?

- House Holder Reduction
- It uses House Holder Rotations (one per column) to reduce a full matrix to a tri-diagonal matrix as we are dealing with symmetric matrices.
- Because, Similarity transformations keep the symmetricy properties of the input matrix.

Final Procedure

1. Reduce the matrix to compact form.
2. Calculate the eigenvalues of compact matrix.
3. Back Transform the eigenvectors.

References

1. http://wwwmayr.in.tum.de/konferenzen/Jass09/courses/2/Kleine_Albers_paper.pdf
2. https://en.wikipedia.org/wiki/QR_decomposition
3. https://en.wikipedia.org/wiki/Givens_rotation
4. http://openmp.org/mp-documents/omp-hands-on-SC08.pdf